### Administrationshandbücher

| **Feld** | **Beschreibung** |
| --- | --- |
| **Name des Artefakts** | Administrationshandbücher |
| **Verantwortlich** | IT-Systemverantwortlicher |
| **Ziel des Dokuments** | Bereitstellung von Anleitungen und Informationen zur Administration und Wartung der Anwendung. |
| **Wesentliche Inhalte** | - Administrationsanleitungen  - Systemkonfigurationsbeschreibungen  - Wartungshandbücher |
| **Erstellungszeitpunkt** | Abschlussphase des Projekts, vor dem Go-Live |
| **Nutzung im Projektverlauf** | Grundlage für die Administration und Wartung der Anwendung |
| **Abnahme** | Durch die IT-Systemverantwortlichen und die Administratoren; Überprüfung der Vollständigkeit und Verständlichkeit der Administrationshandbücher |
| **Abhängigkeiten** | - Systemdokumentationen  - Installationsanleitungen |
| **Versionierung** | Versionskontrolle mittels Versionsmanagement-Tools |
| **Vorlagen und Tools** | Textverarbeitungssoftware (z.B. MS Word, Google Docs) |
| **Zusätzliche Anmerkungen** | Regelmäßige Überprüfung und Aktualisierung der Administrationshandbücher notwendig, um sicherzustellen, dass sie den aktuellen Anforderungen und Implementierungen entsprechen |

**Administrationshandbuch für die CRM-Anwendung**

**1. Einleitung**

Dieses Handbuch bietet eine umfassende Anleitung zur Verwaltung der CRM-Anwendung. Es umfasst Installationsanweisungen, Konfigurationsdetails, Benutzerverwaltung, Datenmanagement und Informationen zur neuen Erweiterung für die Befüllung mit Scoring-Daten.

**2. Systemvoraussetzungen**

* **Betriebssystem**: Linux (Ubuntu 18.04+ empfohlen) oder Windows 10+
* **Datenbank**: MySQL v8 oder höher
* **Node.js**: v14 oder höher
* **RAM**: Mindestens 4 GB
* **Speicherplatz**: Mindestens 20 GB

**3. Installation**

**3.1 Repository klonen**

bash

git clone <repository-url>

cd crm-scoring-system

**3.2 Abhängigkeiten installieren**

bash

npm install

**3.3 Datenbank konfigurieren**

Bearbeiten Sie die config/config.js Datei und fügen Sie Ihre Datenbankdetails hinzu.

**3.4 Datenbank synchronisieren**

bash

npx sequelize-cli db:migrate

**3.5 Server starten**

bash

npm start

**4. Konfiguration**

**4.1 Datenbankeinstellungen**

Die Datenbankeinstellungen befinden sich in config/config.js. Passen Sie diese Datei entsprechend Ihrer Umgebung an.

**4.2 Umgebungsvariablen**

Erstellen Sie eine .env Datei im Stammverzeichnis der Anwendung und fügen Sie die erforderlichen Umgebungsvariablen hinzu:

plaintext

DATABASE\_URL=mysql://<username>:<password>@<host>:<port>/<database>

JWT\_SECRET=<your-secret-key>

**5. Benutzerverwaltung**

**5.1 Benutzer hinzufügen**

Benutzer können über das Admin-Panel oder per API hinzugefügt werden.

**API-Beispiel:**

bash

POST /api/users

Authorization: Bearer <admin-token>

Content-Type: application/json

{

"username": "newuser",

"password": "password",

"role": "admin"

}

**5.2 Benutzerrechte verwalten**

Benutzerrollen und -rechte können über das Admin-Panel oder per API verwaltet werden.

**6. Datenmanagement**

**6.1 Kundendaten importieren**

Kundendaten können über eine CSV-Datei oder direkt über die Datenbank importiert werden.

**6.2 Kundendaten exportieren**

Kundendaten können im Admin-Panel exportiert werden.

**7. Backup und Wiederherstellung**

**7.1 Backup erstellen**

Erstellen Sie regelmäßig Backups der Datenbank:

bash

mysqldump -u <username> -p <database> > backup.sql

**7.2 Backup wiederherstellen**

Stellen Sie ein Backup der Datenbank wieder her:

bash

mysql -u <username> -p <database> < backup.sql

**8. Wartung und Überwachung**

**8.1 Protokolldateien**

Überwachen Sie die Protokolldateien im Verzeichnis /var/log/crm/.

**8.2 Systemüberwachung**

Nutzen Sie Tools wie top, htop, oder systemd für die Systemüberwachung.

**9. Sicherheitsrichtlinien**

**9.1 Benutzerauthentifizierung**

Implementieren Sie eine Zwei-Faktor-Authentifizierung (2FA) für alle Benutzer.

**9.2 Datenverschlüsselung**

Stellen Sie sicher, dass alle Datenübertragungen über HTTPS erfolgen und dass die Datenbank verschlüsselt ist.

**10. Erweiterung um Scoring-Daten**

**10.1 Hochladen von Scoring-Daten**

Benutzer können eine Excel-Datei mit Scoring-Daten über die neue Upload-Seite hochladen. Die Datei wird validiert und die Daten den entsprechenden Kunden zugeordnet.

**10.2 API-Endpunkte**

**10.2.1 POST /upload**

* **Beschreibung**: Hochladen einer Excel-Datei mit Scoring-Daten.
* **Header**: Authorization: Bearer <token>
* **Body**: Multipart-Form-Data mit der Datei.

**10.2.2 GET /customers/:id/score**

* **Beschreibung**: Abrufen des Scoring-Werts eines bestimmten Kunden.
* **Header**: Authorization: Bearer <token>

**10.2.3 PUT /customers/:id/score**

* **Beschreibung**: Manuelles Aktualisieren des Scoring-Werts eines bestimmten Kunden.
* **Header**: Authorization: Bearer <token>
* **Body**: JSON mit dem neuen Scoring-Wert.

**10.3 Fehlerprotokollierung**

Fehlerhafte Datensätze oder Upload-Fehler werden im Fehlerlog protokolliert, das über die Admin-Panel eingesehen werden kann.

**11. Fehlerbehebung**

**11.1 Häufige Probleme**

* **Problem**: Server startet nicht.
  + **Lösung**: Überprüfen Sie die Datenbankverbindung und die Umgebungsvariablen.
* **Problem**: Fehler beim Hochladen der Excel-Datei.
  + **Lösung**: Überprüfen Sie das Dateiformat und stellen Sie sicher, dass alle erforderlichen Felder vorhanden sind.

**11.2 Protokolldateien überprüfen**

Überprüfen Sie die Protokolldateien im Verzeichnis /var/log/crm/ auf Fehlermeldungen und Hinweise.

**12. Kontakt und Support**

Für Fragen und Unterstützung kontaktieren Sie bitte das Entwicklungsteam:

* **Max Mustermann** - *Projektleiter* - max.mustermann@example.com
* **Maria Musterfrau** - *IT-Systemverantwortliche* - maria.musterfrau@example.com